Introspective Classification for Pedestrian Detection

Introspective Classification for Pedestrian Detection

Calum G. Blair and John Thompson
Institute for Digital Communications
University of Edinburgh
Edinburgh, UK
Email: c.blair@ed.ac.uk

Neil M. Robertson
Institute for Sensors, Signals and Systems
Heriot-Watt University
Edinburgh, UK

***Abstract*—State-of-the-art pedestrian detectors are capable of finding humans in images with reasonable accuracy. However, accurate object detectors such as Integral Channel Features (ICF) do not provide good reliability; they are unable to identify detections which they are less confident (or more uncertain) about. We apply existing methods for generating probabilistic measures from classifier scores (such as Platt exponential scaling and Isotonic Regression) and compare these to Gaussian Process classifiers (GPCs), which can provide more informative predictive variance. GPCs are less accurate than ICF classifiers, but GPCs and Adaboost with Platt scaling both provide improved reliability over existing methods.**

I. Introduction

As surveillance and video tracking systems become more common, and the amount of video data they and their human operators process increases, reliance on automated detection of objects within images and videos is similarly becoming more common. Such systems and algorithms must detect relevant objects fast enough to allow timely responses to detections where appropriate, and accurately enough that machine detection does not cause spurious responses, so that operator confidence in system accuracy is maintained.

In this paper, we target a scenario where a detection algorithm is run on video or image data and returns locations of objects of interest, e.g. pedestrians or vehicles [1], [2]. These sliding-window detection algorithms operate on some features $\mathbf{x}$ computed from the video data and produce a vector of window scores $f(\mathbf{x})$, where $f(\cdot)$ denotes the score function. In binary classification tasks, from these the sign $\mathrm{sgn}(f(\mathbf{x}))$ is taken, and windows with positive signs are flagged as containing objects of interest. Given some scoring threshold, a detector may return n window detections above this score. This set may contain false positive and marginal cases along with true positives. These are difficult to separate as the scores from $f(\mathbf{x})$ reflect the interactions between the sample data and the classifier model, rather than classifier confidence in the presence of an object in that window. Finally, this score can be converted to a probability representing confidence in the presence of an object in that location. As an example, see Fig. 1 for pedestrian locations detected by an Adaboost-based classifier.

Most literature on object detection performance has prioritised improvements in *accuracy*; i.e. improvements to the *misclassification* or *error rate*. We define accuracy as *the proportion of samples which are classified correctly*, similarly to Hand [15]. We also define reliability as *how well the classifier's confidence prediction agrees with ground-truth observations*. These two qualities are separate; an accurate classifier which is over- or under-confident will not be reliable. Here, we wish to improve the *reliability* of the confidence measure associated with each detection, so that some action can be taken to reduce uncertainty in the presence or absence of a detection in areas of high uncertainty. This can include applying a more accurate but much more computationally intensive classifier to that region, asking an operator to manually label a limited number of borderline cases, or – in a battlespace scenario – moving a sensor closer to an uncertain region for a closer look. This approach is particularly relevant where the cost of a missed detection can be very high or where the uncertainty-reducing action is too expensive to apply to the image as a whole. Different algorithms can produce different confidence levels from score information. See Fig. 2 for an example of false positive scores.

We use the problem of detecting pedestrians in images as

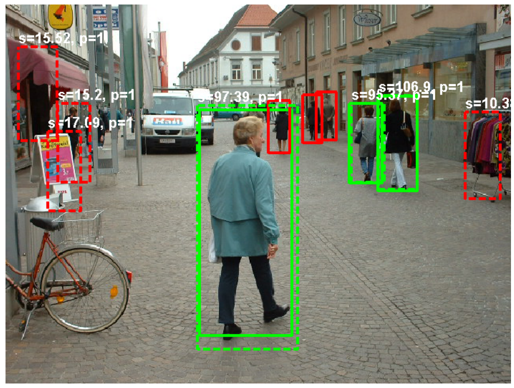
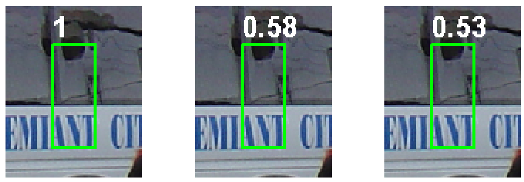


Fig. 1. True positive (dotted green), false positive (dotted red) and false negative (solid red) classifications from the ACF pedestrian detector. Each detection has an Adaboost score s and probability p, which saturates at 1; using this information to rank detections is unreliable.



(a) Sigmoid (b) Platt (c) SE-GP

Fig. 2. Example false positive from Adaboost detector with probabilities shown. Reduced confidence is seen when using Platt or SE-GP postprocessing.

a motivating example in this paper. This is for two reasons: firstly, improved pedestrian detectors can be deployed in tasks which are directly relevant to defence applications and battlespace scenarios (such as anomalous behaviour detection [3], multi-camera surveillance [4], etc.); secondly, techniques used to improve pedestrian detection are applicable to other object classes such as cars and road signs [5], [6]. This analysis technique can also be extended to object detection in other modalities such as sonar or radar imagery.

We evaluate the performance of existing detectors and modify their output to provide more informative confidence levels. We discuss related work then state our contributions in §II. §III describes our method, followed by presentation and discussion of the results in §IV. Finally, a conclusion and avenues of investigation for future work are given in §V.

II. Related Work

A. Pedestrian Detection

A review of the state of the art in pedestrian detection in 2011 is given by Dollàr *et al.* [7], where the authors evaluate sixteen detectors and rank them on multiple pedestrian datasets of varying difficulty. One of the best-performing was based on an extension of the Histograms of Oriented Gradients (HOG) detector [1] called Integral Channel Features (ICF) [8]. Since 2011, performance on the two main datasets, Caltech [7] and INRIA [1], has continued to improve. One of the current state-of-the-art algorithms is Aggregate Channel Features (ACF), a variation of ICF [8]. HOG and its derivatives are sliding window detectors, where window evaluation involves feature extraction at multiple scales followed by classification with support vector machines or boosted decision trees using Adaboost [9]. Individual window detections are then grouped via non-maximal suppression. Improvements in pedestrian detection algorithms can be applied directly to other problems in computer vision. Mathias *et al.* showed that an ICF-based detector can match existing state of the art road sign detectors (achieving 98% accuracy) for a considerable reduction in testing time [5]. Similarly, Rybski *et al.* apply HOG to classifying vehicle orientation in images [6].

B. Classification with Confidence

While the accuracy of pedestrian detectors has continued to improve, algorithm performance at low resolution (less than 50 pixels high) and under varying occlusion is still generally poor [7]. Around half of all detections in the Caltech dataset are still missed at a reasonable rejection level of 10^{-1} false positives per image (FPPI). The presence or absence of detections reported at long-range or in far field image regions is unreliable. A method to generate a reliable confidence score from any detector score is therefore of considerable importance to systems using these detections. Grimmett *et al.* explore this problem in two scenarios, both involving images gathered from a moving vehicle: multi-class road sign classification, and detection of red or green traffic lights [10]. In each case, various approaches are used to generate detection probabilities from classifiers, and it is the evaluation of these methods that concerns us. Grimmett *et al.* use the entropy or uncertainty,

$$H = -\sum_{k=1}^{N} [p(C_k|\mathbf{x})\log_N(p(C_k|\mathbf{x}))] \,, \tag{1}$$

to evaluate the confidence of $N \geq 1$ probabilistic classifiers $p(C_k)$ applied to a window feature vector $\mathbf{x}$. Here, H is bounded between 0 and 1 and a higher H denotes greater uncertainty in a classification. For binary classification $C_k \in \{0, 1\}$, the base 2 logarithm is used.

The authors evaluate two well-known classification algorithms: boosted decision trees (LogitBoost) [11] and support vector machines. These are compared to Gaussian Process classifiers (GP or GPCs) [12]. Having trained these to recognise multiple classes of road signs, they evaluate their ability to identify the presence of *unknown classes*, i.e. those not present in the training set. They find that the classification performance of GPs is similar to that of SVMs and boosted classifiers, but GPs exhibit much more uncertainty than SVMs and LogitBoost when presented with (i) samples of untrained classes, and (ii) false positives and false negative detections of trained classes. They attribute this to the fact that GPs provide probabilistic classification, more so than SVMs or boosted classifiers; the latter do not adequately account for predictive variance [12, Ch. 6]. For example, the SVM output increases in confidence as test points moving further away from the separating hyperplane are selected, when this conclusion may not be supported by the evidence. The major drawback with GPCs is their computational complexity, which is $\mathcal{O}(n^3)$ during training and $\mathcal{O}(n^2)$ at test time [12, Ch. 3]. Grimmett *et al.* do not consider runtime, except to mention that Gaussian Process evaluation is prohibitively expensive.

Other work has evaluated methods for converting classification scores $f(\mathbf{x})$ from SVM or boosted classifiers into probabilities. The simplest of these is a logistic correction via a sigmoid,

$$p(C_k|\mathbf{x}) = \frac{1}{1 + \exp(-2f(\mathbf{x}))} \,. \tag{2}$$

A method involving fitting a model to a given set of data is Platt scaling, originally used with SVMs [13]:

$$p(C_k|\mathbf{x}) = \frac{1}{1 + \exp(-af(\mathbf{x}) + b)} \,. \tag{3}$$

Here a, b are constants learned on a validation set. Yet another involves Isotonic Regression [14], where the output of $f(\mathbf{x})$ is placed into a set of bins. Bin edges and associated probabilities are learned to produce an isotonic (monotonically increasing) mapping to p^*, where $0 \leq p^* \leq 1$.

C. Confidence Metrics

Detector performance is conventionally evaluated based on the four numbers used in a confusion matrix: true positives, false positives, false negative and true negatives (TP, FP, FN and TN respectively). These measures lead to precision, recall, F-score, etc. However, Hand describes two drawbacks of such *error rate*-based statistics for the problem we wish to address [15]. These are: (i) the cost of misclassifying a false positive and a false negative is taken to be equal, and (ii) the relative severity of misclassifications is not taken into account (i.e. an object misclassified just below a threshold will be treated the same as one some distance from it). To overcome this, the mean-squared error or Brier score is used. This relies on the difference between the true class membership $C_k \in \{0,1\}$ and the estimated probability. For binary classification this is:

$$MSE = \frac{2}{N}\sum_{k=1}^{N}(C_k - p(1|\mathbf{x}_i))^2 \,. \quad (4)$$

D. Contributions

We extend the work of Grimmett *et al.* on *classification with confidence*, or *introspective classification*, to the domain of pedestrian detection, by applying and extending their choice of classifiers to the broader feature vector used in FPDW. We also investigate alternative methods for obtaining probabilistic classifications from classifiers running on this dataset, and use a variety of metrics to evaluate each. Here we also consider the tradeoff of algorithm runtime vs. accuracy; i.e. what is an appropriate tradeoff between detector runtime, classifier accuracy and reliability?

III. METHOD

A. Baseline classifier and dataset

We train classifiers on a standard dataset for pedestrian detection, INRIA [1]. For feature extraction we use the baseline detector from ACF[1]. This produces a size 5120-dimension vector for each sliding window, arranged as ten channels comprising L, U, V colour channels, gradient magnitude and gradient orientation histograms with six bins. Each vector represents a 128×64 window. We use eight classifier variations running on this feature vector. As a baseline we use a discrete Adaboost classifier [9], with depth-2 binary trees as weak learners. We also investigate performance of the Logitboost version. As another baseline we use a linear SVM [16] trained on the same feature set[2]. Due to their documented ability to produce probabilistic estimates, we also train a variety of Gaussian Process classifiers on the same feature set. For the classifiers which do not produce probabilistic estimates of test data directly such as Adaboost and SVM, we compare multiple methods of generating probabilistic outputs; see §III-D.

B. Gaussian Processes

For a matrix of training samples X with labels $\mathbf{y}$, test results $\mathbf{f}_*$ can be produced from testing samples X_* by approximating the training and testing distributions via a Gaussian Process [12]:

$$\begin{bmatrix}\mathbf{y}\\ \mathbf{f}_*\end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix}K(X,X) & K(X,X_*)\\ K(X_*,X) & K(X_*,X_*)\end{bmatrix}\right) \,. \quad (5)$$

where the mean is $\mathbf{0}$ and the covariance matrix K can take one of several relationships with the data. For the linear kernel, entry (i,j) of K is computed as:

$$k_{ij}(x) = \sigma_0^2 + \mathbf{x_i}\mathbf{x_j}' \,. \quad (6)$$

Or the squared exponential (SE) kernel:

$$k_{ij}(x) = \exp\left(-\frac{(\mathbf{x_i}-\mathbf{x_j})^2}{2\ell^2}\right) \,. \quad (7)$$

Here, ℓ and σ_0^2 are hyperparameters learned by the training process[3]. Finally, for binary classification a posterior probability $p(f_*|X,\mathbf{y},X_*)$ is obtained from a link function similar to (2) run on (5). A sparse variant of GPCs is the relevance vector machine (RVM), where components of the feature vector in the training data with very low variance are removed during the training process [17]. We compare RVM performance to investigate a GP-based classifier which may have lower computational requirements.

C. Boosted and Margin Classifiers

Adaboost is an algorithm for generating a "well-performing" classifier $f(\mathbf{x})$ from a set of weak learners $h(\mathbf{x})$ [9] using:

$$f(\mathbf{x}) = \sum_{k=1}^{n}(a_k h_k(x)) \,. \quad (8)$$

In this case the weak learners are depth-2 decision trees. We use two variants, discrete Adaboost and Logitboost – the former as it is currently one of the state-of-the-art classifiers and is used in the ACF detector, and the latter as it directly produces a probability estimate via (2). The difference lies in the weight update equation used during training: $D_t(i) \propto \exp(-y_i f_{t-1}(x_i))$ for Adaboost vs. that for Logitboost, $D_t(i) \propto 1/1+\exp(y_i f_{t-1}(x_i))$ [18]. At this stage we also consider the ability of SVMs to produce scores from which probabilistic outputs can be generated. The linear kernel SVM was used to provide a compromise between accuracy and training time:

$$f(\mathbf{x}) = \sum_{i=1}^{N}(\mathbf{x}_i \cdot w_i) + b \,. \quad (9)$$

D. Probabilistic Outputs

Probabilistic classifier outputs are generated either via a non-parametrised sigmoid (2), Platt scaling (3), or isotonic regression. Isotonic Regression uses the pair-adjacent violators algorithm, trained on the scores output by the classifier on a holdout set of training data [14]. A similar method was used to fit Platt parameters on Adaboost scores.

As Gaussian processes produce probabilistic outputs anyway, no further processing is needed. Finally, to investigate a method of generating confidence information with reduced runtime, candidate regions are identified with discrete Adaboost and then these windows are re-processed with the SE GP; we refer to this as Adaboost→SE-GP.

Detector performance is evaluated in the standard manner. Using the INRIA dataset, 288 test images containing one or more pedestrians at varying scales are used. Scores from a sliding window detector are run through non-maximal suppression and compared to a ground truth bounding box corresponding to pedestrian location. If the overlap between the two is over a threshold then the detection is counted as a true positive [7]. See Fig. 1 for examples.

[1]This is available as part of PMT, http://vision.ucsd.edu/~pdollar/toolbox.

[2]The LIBSVM package was used, http://www.csie.ntu.edu.tw/~cjlin/libsvm/.

[3]The GPML toolkit was used. http://www.gaussianprocess.org/gpml/code/matlab/doc/

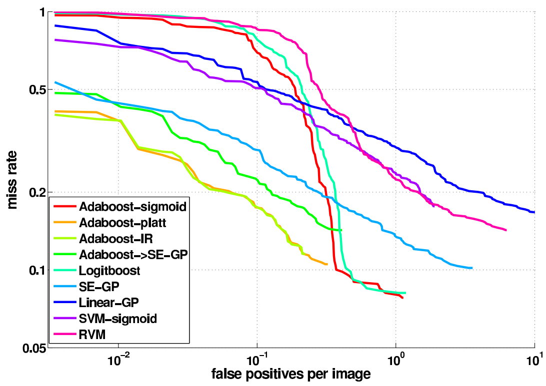


Fig. 3. DET curves for different detectors on INRIA dataset. Best viewed in colour.

IV. Results

As discussed in §II-C, metrics based on the error rate are less applicable to the problem of confidence with classification. They are provided here to allow ranking and comparison of detectors in the usual manner.

A. Misclassification (Inaccuracy) Results

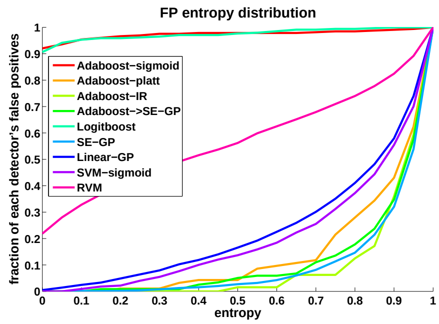
Table I shows results for precision $p = TP/(TP + FP)$, recall $r = TP/(TP + FN)$ and area under ROC curve for each method. Detection Error Tradeoff (DET) curves for probabilistic detectors are given in Fig. 3. Using this metric, the Adaboost classifier (an unmodified ACF detector) with sigmoid or IR fitting performs best. The former has considerable loss of precision at low false-positive rates, despite the underlying detector being the same. However, the SE GPC gives the best AUC despite high false positives.

B. Reliability results

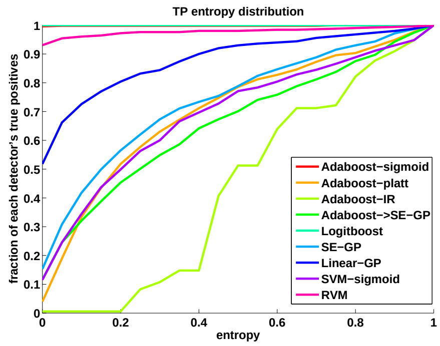
We also evaluate the performance of various approaches for converting a score to a probability. The mean-squared error in the probability score compared to the true $\{0, 1\}$ class for each detector is given in the right-hand side of Table I. Data-driven methods such as Platt scaling and IR outperform simple sigmoid fitting for generating probabilities, and also Adaboost followed by a GPC.

Entropy measures for image regions which returned detections evaluated as false and true positives by each classifier are shown in Fig. 4a and Fig. 4b respectively. These graphs show the distribution of uncertain detections; ideally false positives would be few and uncertain while true positives would have minimal uncertainty (and cluster in the top left corner). Each line is normalised to the number of TP/FP detections returned by that classifier after non-maximal suppression and thresholding. Fig. 4a in particular is affected by the wide variation in number of FPs returned. Adaboost-IR is closest to the ideal detector here, but as Fig. 4b shows, it is *underconfident* when reporting detections. This is relevant if we wish to collect uncertain detections for further processing, e.g. by applying an entropy threshold $H = 0.3$ or $p(1|\mathbf{x}) \simeq 0.91$.

We also display the probability scores using a reliability diagram [14]. Here we wish each curve to be as close to a nominal "well-calibrated" line represented by the diagonal

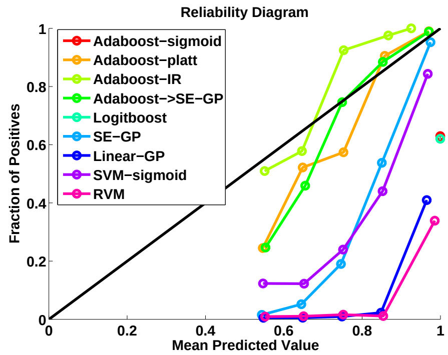


(a) False positives



(b) True positives

Fig. 4. Uncertainty distribution of true and false positive detections for each classifier. The frequency distribution is normalised to the number of true/false detections returned by that classifier. Ideally, most false positive detections would be uncertain, while most true detections would have high probability (low entropy).



1

Fig. 5. Reliability diagram for detection with $p > 0.5$ after NMS. Only bins with >20 detections are shown.

black line as possible. Fig. 5 is built by binning probability values into n bins from 0 to 1 and plotting mean bin values against the mean of the ground truth values for all detections in each bin. To be considered reliable, a detection labelled with 60% confidence should be a true positive 60% of the time. In this case Adaboost→SE-GP and Platt-scaled Adaboost perform well at high probabilities. Fig. 5 only shows detections above a threshold and which have been through non-maximal suppression (NMS) to remove duplicate subwindows. This changes the distribution of detection values. To illustrate the effect of NMS, we also show a reliability diagram in Fig. 6 which uses detection values over all subwindows. Here, the Platt and SE-GP approaches are comparable, especially at

TABLE I. TRUE POSITIVE (TP), FALSE POSITIVE (FP), FALSE NEGATIVE (FN), PRECISION (P), RECALL (R), AREA UNDER CURVE (AUC) AND MEAN-SQUARED ERROR(MSE) OF VARIOUS CLASSIFIERS ON INRIA DATA SET, USING 5120-DIMENSIONAL ACF FEATURE VECTOR.

Name	Probabilistic Correction	TP	FN	FP	p	r	AUC	MSE	runtime(s)
Adaboost	Sigmoid	543	46	326	0.625	**0.922**	0.7596	0.797	0.058
Adaboost	Platt	527	62	93	0.850	0.895	0.8341	**0.331**	0.058
Adaboost	IR	521	68	64	**0.891**	0.885	0.8072	0.346	0.061
Adaboost→SE-GP	N/A	505	84	118	0.811	0.857	0.7960	0.410	N/A
Logitboost	Sigmoid	541	48	341	0.613	0.919	0.7488	0.813	0.060
SE-GP	N/A	529	60	1030	0.339	0.898	**0.8652**	0.595	142
Linear-GP	N/A	500	89	4792	0.094	0.849	0.8062	0.978	120
Linear-SVM	Sigmoid	485	104	548	0.470	0.823	0.7217	0.691	0.45
RVM	N/A	505	84	1811	0.218	0.857	0.7870	1.207	8.683

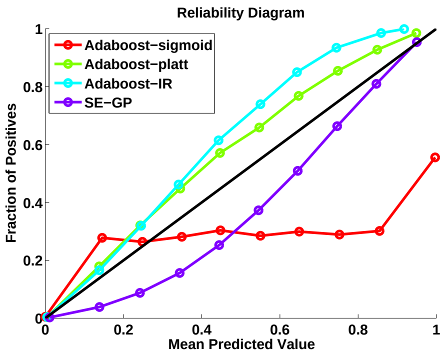


Fig. 6. Reliability diagram for all 'raw' detection scores before NMS and thresholding.

probabilities approaching 1. A decision on whether under- or over-confidence is preferable could be made on an application-specific basis. As Fig. 2 shows, these methods reduce the certainty of false classifications compared to using sigmoids.

C. *Classifier Runtime*

Finally, we also consider detector complexity, and the tradeoff between computation time, accuracy and reliability. As some classifiers are written in C++ and some in MATLAB, a direct comparison is not possible but we list detection runtimes in the last column of Table I. The Adaboost cascade detectors run orders of magnitude faster than GPCs.

V. CONCLUSION

In this work we have extended the analysis of uncertainty in classification as proposed by Grimmett *et al.* [10]. We evaluate various probabilistic approaches and explore how these can be applied to the problem of classification with confidence. Gaussian Process classifiers and Platt-scaled Adaboost both show improved reliability over standard sigmoid methods for indicating uncertain classifications, and are also closer to a well-calibrated detection algorithm. However, for many applications the GPC approach is still too compute- and memory-intensive for use; consequently, better accelerated implementations will be required. Future work will involve applying these classification techniques to other modalities.

ACKNOWLEDGEMENTS

This work was supported by the Engineering and Physical Sciences Research Council (EPSRC) Grant number EP/J015180/1 and the MOD University Defence Research Collaboration in Signal Processing.

REFERENCES

[1] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Comput. Vis. Pattern Recognition, 2005.* IEEE Computer Society, 2005, pp. 886– 893.

[2] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models." *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–45, Sep. 2010.

[3] M. J. Roshtkhari and M. D. Levine, "Online Dominant and Anomalous Behavior Detection in Videos," in *IEEE Conf. Comput. Vis. Pattern Recognit.* Ieee, Jun. 2013, pp. 2611–2618.

[4] R. Vezzani, D. Baltieri, and R. Cucchiara, "People reidentification in surveillance and forensics," *ACM Comput. Surv.*, vol. 46, no. 2, pp. 1–37, Nov. 2013.

[5] M. Mathias, R. Timofte, R. Benenson, and L. Van Gool, "Traffic sign recognition - How far are we from the solution?" in *Int. Jt. Conf. Neural Networks*, Dallas, Aug. 2013.

[6] P. E. P. Rybski, D. Huber, D. D. Morris, and R. Hoffman, "Visual classification of coarse vehicle orientation using Histogram of Oriented Gradients features," in *2010 IEEE Intell. Veh. Symp.* IEEE, Jun. 2010, pp. 921–928.

[7] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian Detection: An Evaluation of the State of the Art," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 4, pp. 743–762, Jul. 2011.

[8] P. Dollar, S. Belongie, and P. Perona, "The Fastest Pedestrian Detector in the West," in *Procedings Br. Mach. Vis. Conf. BMVC 2010*, 2010, pp. 68.1–68.11.

[9] Y. Freund and R. E. Schapire, "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting," *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, Aug. 1997.

[10] H. Grimmett, R. Paul, R. Triebel, and I. Posner, "Knowing When We Don't Know: Introspective Classification for Mission-Critical Decision Making," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2013.

[11] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: a statistical view of boosting," *Ann. Stat.*, vol. 28, no. 2, pp. 337–407, 2000.

[12] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. University Press Group Limited, 2006.

[13] J. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," in *Adv. large margin Classif.*, 1999.

[14] A. Niculescu-Mizil and R. Caruana, "Predicting good probabilities with supervised learning," in *Int. Conf. Mach. Learn.*, no. 1999, 2005.

[15] D. J. Hand, *Construction and Assessment of Classification Rules*. Wiley, 1997.

[16] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1995.

[17] M. Tipping, "Sparse Bayesian learning and the relevance vector machine," *J. Mach. Learn. Res.*, vol. 1, pp. 211—-244, 2001.

[18] R. Schapire, "The boosting approach to machine learning: An overview," *Nonlinear Estim. Classif.*, pp. 1–23, 2003.